R2Q

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/726,144	11/29/2000	Frank Worrell	00-419 1496.00055	6557

24319 7590 08/13/2004

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106 LEGAL
MILPITAS, CA 95035

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/726,144	WORRELL, FRANK	
	Examiner	Art Unit	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2004 and 19 July 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14 and 16-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14 and 16-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-14 and 16-20 have been examined.

Papers Received

2. Receipt is acknowledged of the request for continued examination (RCE) paper, where the paper has been placed of record in the file.

Response to Amendment

3. The argument for the 35 USC 132 objection of new matter is persuasive and thus this objection as well as the drawing objection and 35 USC 112 rejections have been withdrawn (note that new 35 USC 112 rejections have been added). However, the examiner requests that figure 3, be amended to place dashed lines between the different stages so it can be easily seen which blocks belong to which stages. This will allow for the specification to clearly show that a branch target is fetched at the same time that a sequential instruction is decoded.

Claim Rejections - 35 USC § 112

4. Claims 1-14 and 16-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. The term "substantially simultaneously" in claims 1, 2, and 7 is a relative term which renders the claim indefinite. The term "substantially simultaneously" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. The word “simultaneous” means that multiple events occur

at the same exact time, or in this case perhaps during the same pipeline stage or clock cycle. There is no relative degree of the word simultaneously; either the events happen simultaneously or they do not. The examiner is taking the claims to mean "simultaneously" rather than "substantially simultaneously" since the Applicant is trying to show simultaneous events as described in the argument against the 35 USC 132 objection.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-2, 14, and 16 are rejected under 35 U.S.C. 102(b) as being anticipated by Trauben (5,509,130).

8. In regard to claim 1, Trauben discloses a method of conditional branching in a pipelined processor (figure 9 gives the flow of the pipeline stages), the method comprising the steps of:

a. fetching a first instruction stored at a branch target address in response to encountering a branch instruction, at a program counter address; [Figure 10b shows that when a branch instruction (BNE) is encountered or decoded (d0 and d1), the first Target is fetched (T1). Figure 1, element 12 shows that a program counter exists in the system. Column 5, lines 35-46 show that the program

counter is used for issuing instructions and its function does not need to be described because it is well know in he art. This well known in the art function is that the program counter points to the addresses of instructions that are fetched for execution.]

b. decoding a second instruction stored at a next address adjacent said program counter address simultaneously with said fetching; [Figure 10b shows that a delay instruction and the sequential instruction after it are decoded while fetching T1. Column 18, lines 51-60 further explain that these are sequential instructions with respect to the branch (because of their placement in the sequential queue) that decode while the target is fetched. Since these instructions are sequential, they are adjacent to said program counter address of the branch.]

c. evaluating between taking a branch defined by said branch instruction and not taking said branch simultaneously with said fetching. [Column 18, line 61 shows that at time 6, the branch is resolved, meaning that it was being evaluated during time 5, or while the fetching was occurring. Figure 10b further shows with arrows that the execute stage e0 determines the condition of the branch so the next instructions to be decoded are known and that this evaluation occurs at the same time as the fetching of the target T1.]

d. And fetching a third instruction stored at a mispredict recovery address adjacent said next address in response to determining not to take said branch. [Figure 10b shows the case where the branch is not taken (column 18, lines 41-

42). As shown above, the next address points to the Delay and Sequential 1 instructions. The term adjacent does not define the address immediately sequential to the "next address" but simply an address near it. Therefor, adjacent to the "next address" is S4 (Sequential 4), which is shown in the figure to be fetched (with Sequential 3) in response to the resolution of the branch (e0); that is in determining to not take it. Thus, a third and fourth instruction are fetched. This is backed up with the description in column 19, lines 9-39.]

9. In regard to claim 2, Trauben discloses a method of conditional branching in a pipelined processor (figure 9 gives the flow of the pipeline stages), the method comprising the steps of:

a. fetching a first instruction stored at a branch target address in response to encountering a branch instruction, at a program counter address; [Figure 10a shows that when a branch instruction (BNE) is encountered or decoded (d0 and d1), the first Target is fetched (T1). Figure 1, element 12 shows that a program counter exists in the system. Column 5, lines 35-46 show that the program counter is used for issuing instructions and its function does not need to be described because it is well know in he art. This well known in the art function is that the program counter points to the addresses of instructions that are fetched for execution.]

b. decoding a second instruction stored at a next address adjacent said program counter address simultaneously with said fetching; [Figure 10a shows that a delay instruction and the sequential instruction after it are decoded while

fetching T1. Column 18, lines 51-60 further explain that these are sequential instructions with respect to the branch (because of their placement in the sequential queue) that decode while the target is fetched. Since these instructions are sequential, they are adjacent to said program counter address of the branch.]

c. evaluating between taking a branch defined by said branch instruction and not taking said branch simultaneously with said fetching. [Column 18, line 61 shows that at time 6, the branch is resolved, meaning that it was being evaluated during time 5, or while the fetching was occurring. Figure 10a further shows with arrows that the execute stage e0 determines the condition of the branch so the next instructions to be decoded are known and that this evaluation occurs at the same time as the fetching of the target T1.]

d. And fetching a third instruction stored at a sequential instruction address adjacent said branch target address in response to determining to take said branch. [Figure 10a shows the case where the branch is taken (column 18, lines 40-41). As shown above, the next address points to the Delay and Sequential1 instructions of figure 10b that are decoded while the target T1 is fetched. The term adjacent does not define the address immediately sequential to the "branch target address" but simply an address near it. Therefor, adjacent to the "branch target address" is T3 (Target 3, a sequential instruction following the target instruction), which is shown in the figure to be fetched in response to the

resolution of the branch (e0); that is, in determining to not take it. This is backed up with the description in column 18, line 43 – column 19, line 8.]

10. In regard to claim 14, language similar to that of claim 1 is provided and as such, the same arguments given above to claim 1 apply to claim 14.

11. In regard to claim 16, Trauben discloses the method of claim 1, further comprising the step of: storing said program counter address in a stage of said pipelined processor for at least two cycles. Since instructions are fetched each stage (figure 10), the program counter address is used each stage for this fetching and thus the program counter address is stored for at least two cycles. Though the program counter address may have a different value, it is still the program counter address nonetheless that is stored in the program counter each cycle.

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 3-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Trauben in view of Hennessy.

14. In regard to claim 3,

a. Trauben discloses the method of claim 2;

b. Trauben does not explicitly disclose further comprising the step of: generating said sequential instruction address based upon said program counter address and a predetermined offset.

c. However, Trauben has shown in column 5, lines 35 – 46 that a program counter is used and the program counter is intended to be among a broad category of program counters found in most pipelined processors. Hennessy has disclosed on pages 385 and 404 the use of a program counter that is incremented by a predetermined offset to generate the next sequential instruction address.

d. Since Trauben has taught that any program counter may be used one of ordinary skill in the art would have been motivated to modify the design of Trauben to use the program counter and update method taught by Hennessy. Trauben has taught that the system of use involves fixed length instructions (column 2, lines 54-62) and thus the program counter incrementation method would work successfully to address the sequential instruction address, which may be the address of a group of two sequential instructions (figure 10).

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben to us the program counter and method for updating taught by Hennessy since Trauben discloses that any program counter among a broad category of common program counters may be used.

15. In regard to claim 4,

a. Trauben discloses the method of claim 1 including the use of a misprediction recovery address based upon an exception program counter address; [As shown in figure 10b and column 19, lines 9-40, a sequential address is generated, and the instruction there is fetched. This instruction is executed when the branch is incorrectly predicted and is then a misprediction recovery instruction at a misprediction recovery address. As shown above, all instructions are pointed to by a program counter.]

b. Trauben does not disclose further comprising the step of: generating said misprediction recovery address based upon an exception program counter address and a predetermined offset.

c. However, Trauben has shown in column 5, lines 35 – 46 that a program counter is used and the program counter is intended to be among a broad category of program counters found in most pipelined processors. Hennessy has disclosed on pages 385 and 404 the use of a program counter that is incremented by a predetermined offset to generate the next sequential instruction address.

d. Since Trauben has taught that any program counter may be used one of ordinary skill in the art would have been motivated to modify the design of Trauben to use the program counter and update method taught by Hennessy. Trauben has taught that the system of use involves fixed length instructions (column 2, lines 54-62) and thus the program counter incrementation method

would work successfully to address the sequential instruction address, which may be the address of a group of two sequential instructions (figure 10).

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben to us the program counter and method for updating taught by Hennessy since Trauben discloses that any program counter among a broad category of common program counters may be used.

16. In regard to claim 5, Trauben discloses the method of claim 1, as described above, further comprising the step of:

a. generating said branch target address; Trauben shows in column lines 16-19 that a branch target address is calculated or generated. Trauben also discloses the existence of a program counter as shown above.

b. Trauben does not disclose that this branch target address is based upon a program counter address and an address displacement of said branch instruction.

c. Hennessy, on page 148 discloses the use of an offset or displacement to be added to some position stored in a register, which yields a branch target address. The program counter given by Trauben is a register because it holds a value in close proximity as do general-purpose registers.

d. Hennessy discloses that using his method to calculate the branch target address provides a large range of possible targets for the branch. Hennessy also teaches on last paragraph that by using the PC as the register, hardware convenience is established. This large range of addressing possibilities and

hardware convenience taught by Hennessy would have motivate done of ordinary skill in the art to implement the method of Hennessy in Trauben for greater address range and hardware convenience.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify Trauben's invention to generate the branch target address based upon a program counter address and an address displacement of the branch instruction as taught by Hennessy in order to have a wide range of addresses (also taught by Hennessy) to branch to and for hardware convenience.

17. In regard to claim 6, similar language is used (in part) to that of claim 2 concerning the fetching of an instruction in response to determining not to take said branch and thus the same arguments provided above for claim 2 apply to this portion of claim 6. In addition, the use of updating a program counter and an exception program counter is disclosed above for claims 3 and 4 and thus the same arguments apply here. Concerning the limitation regarding generating the branch target address based on a displacement, Trauben shows in column 18, lines 16-19 that the branch target address is computed. Trauben also discloses the existence of a program counter as shown above. Since the target address must be computed, an absolute address cannot be taken. Thus it is inherent that the branch address is computed using the current address (program counter) and some displacement pointing to the next instruction.

18. Claims 7-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Trauben in view of Kahle (6,678,820).

19. In regard to claim 7,

a. Trauben discloses a pipelined processor (figure 1) with a branch target address (figures 10a and 10b, Target 1 or T1) based on a branch instruction stored at a program counter and a sequential instruction address (figures 10a and 10b, Sequential1 or S1) having a first value adjacent said program counter address; [Figure 1, element 12 shows that a program counter exists in the system. Column 5, lines 35-46 show that the program counter is used for issuing instructions and thus the branch target address, which points to a branch instruction to be issued is based on the program counter. Figures 10a and 10b shows that the sequential instruction is adjacent (or near), in program order, the target instruction and thus the addresses are adjacent since the addresses specify program order.]

b. Trauben does not disclose the processor comprising: a multiplexer and a circuit configured to present (i) a branch target address based on a branch instruction stored at a program counter address in prediction of taking a branch, (ii) a sequential instruction address having a first value adjacent said program counter address, (iii) and a mispredict recovery address to said multiplexer simultaneously.

c. Kahle has disclosed a multiplexer (figure 2, element 126) and a circuit configured to present (i) a branch target address, (ii) a sequential instruction address, (iii) and a mispredict recovery address to said multiplexer simultaneously. [Column 5, line 54 – column 6, line 7 of Kahle show that addresses of branch instructions are predicted and that a target address and

sequential instruction address are presented to the multiplexer 126. As discussed previously, the target address and sequential address both serve as mispredict recovery addresses. In the case that the branch target is the selected path (as described in Kahle) and this prediction is incorrect, the sequential address is inherently the recovery address since there are only two possible paths, target and sequential. Likewise if the sequential is predicted and incorrect, the target address is the recovery address. Thus the target address, sequential address, and recovery address are all input to the multiplexer simultaneously.]

d. Column 5, lines 54-56 show that the branch prediction of Kahle includes both address and direction prediction. Column 1, line 60 – column 2, line 15 show that branch prediction minimizes execution stalls by allowing instructions to be fetched prior to branch resolution. By using two levels of branch prediction (address prediction in addition to direction prediction), one of ordinary skill in the art would recognize that stalls can be better avoided by having branch information including the predicted address ready before branch resolution. Trauben does not use address prediction but instead simply fetches both target and sequential instructions and selects the appropriate instruction path based on the branch resolution or prediction as shown in figure 3 and column 8, line 37 – column 9, line 5. The ability to use address prediction so that stalls may be minimized as taught by Kahle would have motivated one of ordinary skill in the

art to modify the design of Trauben to use the address prediction method disclosed by Trauben.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben to use the address prediction scheme disclosed by Kahle so that execution stalls are minimized as taught by Kahle.

20. In regard to claim 8, Trauben in view of Kahle discloses the pipelined processor of claim 7, as described above, wherein said circuit is further configured to present said sequential instruction address having a second value adjacent said branch target address to said multiplexer in response to determining to take said branch. [The claim is referring to a second sequential value, which will be interpreted as a Target 3 or T3 instruction as given in figure 10a and column 18, line 43 – column 19, line 8. When the branch is predicted taken, T3 is fetched, and thus provided to the multiplexer. This instruction is adjacent or near the target (T1) instruction and is sequential to the target instruction.]

21. In regard to claim 9, Trauben in view of Kahle discloses the pipelined processor of claim 7 wherein said circuit comprises: a prefetch program counter for storing an address presented by said multiplexer among said branch target address, said sequential instruction address and said misdirect recovery address. [Figure 2 of Kahle shows that a register 142is used to store the address presented by the multiplexer and thus is being interpreted as a prefetch program counter.]

22. In regard to claim 10, Trauben in view of Kahle discloses the pipelined processor of claim 7 wherein said circuit comprises an instruction register for storing said branch

instruction. The instruction cache 34 of figure 3 stores the instructions and thus also the branch instruction. The enclosed IEEE definition of "register" states that a register is a device capable of retaining information of the aggregate information in a digital computer and thus the instruction cache is an instruction register.

23. In regard to claim 11, Trauben in view of Kahle discloses the pipelined processor of claim 7, wherein said circuit comprises: an exception program counter for storing an exception program counter address used upon determining to not take said branch. As shown above, when the branch is determined as not taken, a mispredict recovery address is fetched from. It has been shown to be the sequential instruction after the branch instruction. Thus the program counter, which points to this instruction, is also an exception program counter that stores an exception program counter address. Though Trauben in view of Kahle does not explicitly state that the exception program counter is disposed in the decode stage, it is shown to have the same claimed functionality. The shifting of the exception program counter part to the decode stage to perform the same function provides no new or unexpected result over the prior art. Therefore, one of ordinary skill in the art would have found it obvious to shift the exception program counter to the decode stage of the pipeline in Trauben in view of Kahle and from there perform the functionality given above (see In re Japikse 86 USPQ 70 (CCPA 1950)).

24. Claims 12-13 and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Trauben in view of Kahle as applied to claim 7 above, and further in view of Hennessy.

25. In regard to claim 12,

a. Trauben in view of Kahle discloses the pipelined process of claim 7 wherein said circuit comprises: an exception program counter for storing an exception program counter address used upon determining not to take said branch. As shown above, when the branch is determined as not taken, a mispredict recovery address is fetched from. It has been shown to be the sequential instruction immediately after the branch instruction. Thus the program counter, which points to this instruction, is also an exception program counter that stores an exception program counter address.

b. Trauben in view of Kahle does not disclose that the exception program counter is disposed in an execution stage of said pipelined processor.

c. Hennessy has taught on page 404, top paragraph, that a sequential PC is always computed for every instruction. The mispredict recovery address is simply a sequential address from the branch instruction address. On the table directly below, Hennessy has taught that the sequential instruction address is generated using two source operands, one being the PC (program counter). Hennessy shows on page 506 the use of an exception program counter (EPC) disposed in the execute stage of the pipelined processor.

d. By always computing the sequential address from the branch instruction address in the PC using the method described by Hennessy, a uniform and simple method of sending control to the next instruction is established. Hennessy teaches on page 509 that the exception (a mispredicted branch in this case) is detected in the decode stage and handled in the execute stage. By

handling it in the next clock cycle, one can be assured that the correct information will be present. This uniformity and simplicity coupled with integrity of data would have motivated one of ordinary skill in the art to modify Trauben in view of Kahle's design to incorporate the method of generating misprediction recovery address given by Hennessy.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben in view of Kahle to include the sequential addressing method of Hennessy in order to have a uniform and simplistic design with a high integrity of data while generating a misprediction recovery address.

26. In regard to claim 13,

a. Trauben in view of Kahle discloses the pipelined processor of claim 7, wherein said circuit comprises:

i. a prefetch program counter for storing said program counter address; Figure 3, shows that the sequential queue stores or prefetches the sequential instructions received from the instruction cache. The program counter in the figure is shown to address these prefetched instructions and is thus the prefetch program counter.

ii. an instruction register for storing said branch instruction; The instruction cache 34 of figure 3 stores the instructions and thus also the branch instruction. The enclosed IEEE definition of "register" states that a register is a device capable of retaining information of the aggregate

information in a digital computer and thus the instruction cache is an instruction register.

b. Trauben in view of Kahle does not disclose an exception program counter for storing an exception program counter address used in generating said mispredict recovery address having a second value proximate said program counter.

c. Hennessy has taught on page 404, top paragraph, that a sequential PC is always computed for every instruction. The mispredict recovery address is simply a sequential address from the branch instruction address. On the table directly below, Hennessy has taught that the sequential instruction address is generated using two source operands, one being the PC (program counter). Hennessy shows on page 506 the use of an exception program counter (EPC) disposed in the execute stage of the pipelined processor that has a second value inherently proximate (in the vicinity) of the program counter.

d. By always computing the sequential address from the branch instruction address in the PC using the method described by Hennessy, one of ordinary skill in the art would recognize that a uniform and simple method of sending control to the next instruction is established. Hennessy teaches on page 509 that the exception (a mispredicted branch in this case) is detected in the decode stage and handled in the execute stage. By handling it in the next clock cycle, one of ordinary skill in the art can be assured that the correct information will be present. This uniformity and simplicity coupled with integrity of data would have motivated

one of ordinary skill in the art to modify Trauben in view of Kahle's design to incorporate the method of generating a misprediction recovery address given by Hennessy. As it relates to Trauben in view of Kahle, the exception (misprediction) would be detected in the decode stage of the branch (BNE) instruction of figure 10b and handled in the execute stage.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben in view of Kahle to include the sequential addressing method of Hennessy in order to have a uniform and simplistic design with a high integrity of data while generating a misprediction recovery address.

27. In regard to claim 17, Trauben in view of Kahle and Hennessy discloses the pipelined processor of claim 13, wherein said circuit further comprises: an incrementor coupled to said prefetch program counter address and configured to generate said sequential instruction address from aid program counter address. As shown above and the prefetch program counter address is stored in the program counter. Since the program counter is updated or incremented by a predetermined value as taught in Hennessy on page 404, it is inherent that an incrementor exists to do this increment and update the program counter to the next sequential instruction. This is supported by Trauben who uses fixed-length instructions as set forth in column 2.

28. In regard to claim 18, Trauben in view of Kahle and Hennessy discloses the pipelined processor of claim 13, wherein said circuit further comprises: an adder (i) coupled to both said prefetch program counter and said instruction register and (ii) configured to generate said branch target address by adding said program counter

address to an address displacement of said branch instruction. Column 18, lines 16-19 of Trauben show that the branch target address is computed. Since the branch target addresses are computed and an absolute address is not taken, it is inherent that this address is generated from the current address (stored in the program counter) and an offset (displacement) that when added to the current address gives the destination as described above for claim 5.

29. In regard to claim 19, Trauben in view of Kahle and Hennessy discloses the pipelined processor claim 13, wherein said circuit further comprises: an incrementor coupled to said exception program counter and configured to generate said mispredict recovery address. It is inherent that an incrementor exists to update the program counters (including the exception program counter) to the next sequential instruction as presented above for claim 17.

30. In regard to claim 20, Trauben in view of Kahle and Hennessy discloses the pipelined processor of claim 13, wherein said exception program counter is coupled to said prefetch program counter to receive said program counter address. As shown in the caption on page 506 of Hennessy, the exception program counter saves the address of the instruction that caused the exception. Since the prefetch program counter contains the current instruction address, when the branch instruction generates the exception, the exception program counter is inherently updated from the prefetch program counter.

Response to Arguments

31. Applicant's arguments with respect to the suggested title have been fully considered and are persuasive.

32. Applicant's arguments filed 26 May 2004 in regards to claims 1, 2, 5, 14, and 16 have been fully considered but they are not persuasive.

33. In regard to claims 1 and 14, Applicant is concerned that Trauben does not teach fetching a sequential instruction after evaluation a branch but instead does so beforehand. Figure 10b shows the case where the branch is not taken (column 18, lines 41-42). As shown above, the next address points to the Delay and Sequential 1 instructions. The term adjacent in the claim does not define the address immediately sequential to the "next address" but simply an address near it. Therefor, adjacent to the "next address" is S4 (Sequential 4), which is shown in the figure to be fetched (with Sequential 3) in response to the resolution of the branch (e0); that is in determining to not take it. Thus, a third and fourth instruction are fetched. This is backed up with the description in column 19, lines 9-39.

34. In regard to claim 2, Applicant is concerned that Trauben does not teach fetching a target instruction after evaluation a branch but instead does so beforehand. Figure 10a shows the case where the branch is taken (column 18, lines 40-41). As shown above, the next address points to the Delay and Sequential1 instructions of figure 10b that are decoded while the target T1 is fetched. The term adjacent does not define the address immediately sequential to the "branch target address" but simply an address near it. Therefor, adjacent to the "branch target address" is T3 (Target 3, a sequential instruction following the target instruction), which is shown in the figure to be fetched in

response to the resolution of the branch (e0); that is, in determining to not take it. This is backed up with the description in column 18, line 43 – column 19, line 8.

35. In regard to claim 5, Applicant has argued that claim 5 does not provide that the motivation is drawn from a reference or knowledge to one of ordinary skill in the art. However, as shown in the previous argument, clear indication has been given that Hennessy teaches the motivation since sections of Hennessy are referred to as teaching the subject matter used for motivation. Further emphasis has been added for clarification.

36. In regard to claim 16, Applicant argues the program counter does not hold the value for at least two cycles. Since instructions are fetched each stage (figure 10), the program counter address is used each stage for this fetching and thus the program counter address is stored for at least two cycles. Though the program counter address may have a different value, it is still the program counter address nonetheless that is stored in the program counter each cycle.

37. Applicant's arguments, see arguments, filed 26 May 2004, with respect to the rejection(s)of claim(s) 3, 4, and 6 under Trauben have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Hennessy.

38. The argument for claim 3 that asserts Trauben uses variable length instructions is not believed to be true. The section referred to by the Applicant (column 2, lines 43-45) indicate a traditional variable length instruction set, however, column 2, line 54 – column 3, line 9 show that fixed length instructions are desirable and a part of the

disclosed invention. The examiner has, however, used a new grounds of rejection in order to rid any doubt over inherency and explicitly show the use of adding predetermined values to a program counter to address instructions. The same argument was made for claims 4 and 6, which has been partially responded to above, and as with claim 3 a new grounds of rejection has been made specifically showing the use of a program counter that is updated with predetermined values to address instructions.

39. In regard to claim 6, applicant admits that the same issue is involved as for claim 2 and thus the argument given above for claim 2 applies.

40. Applicant's arguments, see arguments, filed 26 May 2004, with respect to the rejection(s)of claim(s) 7-13 and 17-20 under Trauben and Trauben in view of Hennessy have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Kahle.

41. In regard to claim 12, clear indication that the motivation is from Hennessy has been given as argued above. Regarding the reasonable expectation of success, applicant states that Hennessy teaches a sequential PC is always computed for every instruction, and asserts this to conflict with the teaching of Trauben and the ability therein to issue multiple instructions simultaneously. The examiner would like to point out that a PC address points to each fetched group and thus each group has a computed sequential PC for it and thus each instruction in that group has a sequential PC (though the same PC for each instruction of the group) computed for it.

42. In regard to claim 13, Applicant states that the reasonable expectation of success is flawed. The examiner points out that as it relates to Trauben, the decoding and detecting of the misprediction or exception occurs in the decode stage of the branch instruction and not the branch target instruction.

43. The arguments for the other claims of this group have been given new arguments as shown above in order to answer the Applicant's remarks and meet the deficiencies stated therein.

Conclusion

44. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

45. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The cited references given in the previous Office action remain pertinent and are cited with this action as well.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
August 9, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100